DOCUMENT RESUME

ED 232 688 IR 050 336

AUTHOR Declerck, Luc; Craven, Timothy C.
TITLE NETPAD User Manual.
INSTITUTION University of Western Ontario, London. School of Library and Information Science.
SPONS AGENCY Natural Sciences and Engineering Research Council, Ottawa (Ontario).
PUB DATE 83
GRANT A3027
NOTE 80p.
PUB TYPE Guides - General (050)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS *Computer Programs; *Display Systems; Guidelines; *Indexing; Information Retrieval; *Man Machine Systems; *Microcomputers; Online Systems; *Permuted Indexes; Subject Index Terms

ABSTRACT

NETPAD is a microcomputer-based system, implemented on a Commodore PET 2001-8, by which a human indexer can input, retrieve, and edit information in order to produce a variety of permuted index displays. The basic unit of the NETPAD system is a subnetwork (a description of a single document) containing a set of concept nodes (author, title, date, subject term(s), etc.) and a set of links (conceptual relations) between concept nodes. This manual has been prepared as a self-instructing tool for beginning or experienced indexers, NETPAD users, and researchers. It is noted that previous training in library science or indexing is not necessary to use the manual. Following introductory material, the manual covers: (1) the equipment needed to use NETPAD; (2) start-up procedures; (3) system commands; (4) inputting, editing, and sorting subnetworks; (5) displaying subnetworks in table form; (6) storing subnetworks; (7) clearing the workspace; (8) retrieving subnetworks; (9) deleting subnetworks; (10) generating permuted index entries; (11) customizing permuted index displays; (12) maintaining the inverted index file; (13) producing hard copies; (14) standardizing index terms; (15) generating facet printouts; (16) using index productivity aids; and (17) exiting from NETPAD. A summary of commands concludes the manual. (ESR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED232688

NETPAD USER MANUAL

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

LUC DECLERCK

AND

TIMOTHY C. CRAVEN

SCHOOL OF LIBRARY AND INFORMATION SCIENCE
THE UNIVERSITY OF WESTERN ONTARIO
LONDON, ONTARIO
N6G 1H1

1983

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

T.C. Craven

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR050336

ERIC Clearinghouse on
Information Resources
Syracuse University
School of Education
Syracuse, N. Y. 13210
(315) 423-3640

AUTHOR ABSTRACT
(To Accompany Document Submitted to ERIC/IR)

Author: Luc Leclerck and Timothy C. Craven

Title: NETPAD User Manual

Abstract (up to 200 words):

This brief manual has been prepared as a self-instructing tool for anyone interested in indexing documents using the NETPAD system or in experimenting with it for other reasons. Readers may be beginning or experienced indexers, NETPAD users, or researchers interested in further work on the NETPAD system. Previous training in library science or indexing has not been assumed.

NETPAD is a microcomputer-based system by which a human indexer can input, retrieve, and edit in graphic form networks of concepts and concept links from which a variety of permuted index displays can be produced on demand.

The system has been implemented on a PET2001-8 microcomputer. All the program modules of the system are stored on one diskette of a CBM 2040 dual floppy diskette drive, and the database from which the index displays are produced is stored on the other.

****
Information About Document
****

Paper presented at:

Project Report (Contract Information, if possible)

Journal Article (Complete Citation)--(Some journals do not appear in CIJE)

Other User manual

052326

ACKNOWLEDGEMENTS

The writing of this manual was supported in part by Individual Operating Grant A3027 of the Natural Sciences and Engineering Research Council of Canada.

SOFTWARE AVAILABILITY

To obtain a copy of the NETPAD software in its current form, please send one soft-sectored 5 1/4 inch floppy diskette, such as a Scotch 744-0, to:

Timothy C. Craven,
School of Library and Information Science,
Elborn College,
The University of Western Ontario,
London, Ontario N6G 1H1
Canada.

TABLE OF CONTENTS

1.0 Introduction 5

1.1 Purpose of the NETPAD manual 5
1.2 What is NETPAD? 5

2.0 Getting started 6

2.1 Equipment needed 6
2.2 Start-up procedures 6
2.3 System commands 9

3.0 Subnetworks 10

3.1 Introduction 10
3.2 Concept nodes 10

3.2.1 Inputting concept nodes 11
3.2.2 Editing concept nodes 14
3.2.3 Deleting concept nodes 16

3.3 Links 18

3.3.1 Defining linktypes 18
3.3.2 Inputting linktypes 20
3.3.3 Editing linktypes 24
3.3.4 Assigning links between concept nodes 24
3.3.5 Unlinking concept nodes 27

3.4 Appending concept nodes and links 28
3.5 Sorting subnetworks 30

4.0 Displaying subnetworks in table form 32

5.0 Storing subnetworks 33

6.0 Clearing the workspace 34

7.0 Retrieving subnetworks 35

7.1 Retrieving subnetworks via the QUERYING module .. 35
7.2 Retrieving subnetworks via the BROWSING module .. 39

8.0 Deleting subnetworks 50

9.0 Generating permuted index entries 51

10.0 Customizing permuted index displays 53

10.1 Specification of different citation orders 53
10.2 Exclusion of selected item descriptions 56
10.3 Controlling heading boundaries 57
10.4 Rewording connective expressions 58
10.5 Resetting weights and connectives 59

10.6 Setting subheading and cutoff thresholds 59

11.0 Maintaining the inverted-index file 60

12.0 Producing hard copies 62

13.0 Standardizing index terms 63

14.0 Generating facet printouts 65

15.0 Using indexer productivity aids 67

15.1 Syntactic analysis of phrases 67
15.2 Automatic extraction of index terms 69
15.3 Using subschemas 72

16.0 Exiting from NETPAD 73

17.0 Summary of commands..... 74

17.1 General commands 74
17.2 Module-bound commands 75
17.3 Additional commands 78

1.0 INTRODUCTION

1.1 Purpose Of The NETPAD Manual

This brief manual has been prepared as a self-instructing tool for anyone interested in indexing documents using the NETPAD system or in experimenting with it for other reasons. Readers may be beginning or experienced indexers, NETPAD users, or researchers interested in further work on the NETPAD system. Previous training in library science or indexing has not been assumed.

1.2 What Is NETPAD?

NETPAD is a microcomputer-based system by which a human indexer can input, retrieve, and edit in graphic form networks of concepts and concept links from which a variety of permuted index displays can be produced on demand.

The system has been implemented on a PET2001-8 microcomputer. All the program modules of the system are stored on one diskette of a CBM 2040 dual floppy diskette drive, and the database from which the index displays are produced is stored on the other.

2.0 GETTING STARTED

2.1 Equipment Needed

The minimum amount of equipment needed to run the NETPAD system is as follows:

1. a Commodore PET microcomputer, Series 2001, with version 2 of BASIC in ROM,
2. a Commodore Dual Drive Disk Unit, Model 2040,
3. a PET-to-IEEE cable,
4. a NETPAD package diskette, 5 1/4 inch,
5. a blank, an unused, or an existing NETPAD database diskette, 5 1/4 inch.

Note that the diskettes (also known as floppy disks, minidisks, etc.) should be soft-sectored.

If hard copies are desired a Commodore CBM Printer and an IEEE-488 cable must be added.

2.2 Start-up Procedures

There are seven basic steps involved in starting-up the NETPAD system. They are as follows:

1. Switch the microcomputer and disk drives on
2. Insert the NETPAD package diskette in disk drive 0
3. Type: OPEN15,8,15 (plus RETURN)

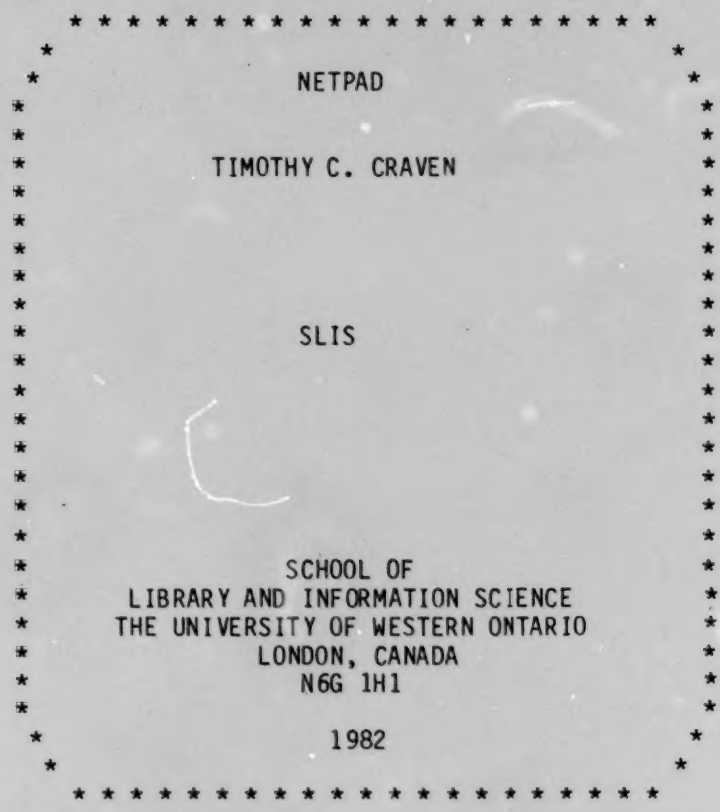
4. Type: PRINT#15,"IO" (plus RETURN)

5. Type: LOAD"0:NETPAD",8 (plus RETURN)

6. Insert a diskette with a NETPAD database on, or alternatively an unused or erased diskette on which you want to create a NETPAD database, in disk drive 1 (The easiest way of erasing a diskette is to expose it to a magnetic field. This can be done by running a small magnet over the envelope's surface.)

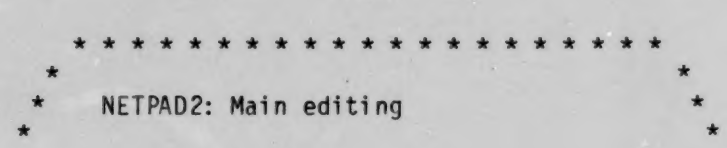
7. Type: RUN (plus RETURN)

At that point, the system will automatically respond with the following display:



This display however will only appear on the screen for a few seconds, following which the system will check whether the diskette inserted in disk drive 1 contains a database or is blank.

If the diskette contains a database, the system will automatically assume that it is the right one and pass system command to NETPAD's default module which will in turn open the keyboard for input and display the module heading in the top left corner of the screen; as follows:



On the other hand, if the system detects a blank diskette in disk drive 1, it will verify the user's intention by displaying the following prompt:

NEW DATABASE?

If the user responds with the letter "N", the system will generate the following prompt:

DATA DISK PROPERLY INSERTED?

If at that stage, the user types the letter "Y" the system will re-attempt to accept the existing database. If the user types "N" the system will abort and the command "RUN (plus RETURN)" will need to be typed in to re-activate the system.

If following the prompt "NEW DATABASE?", the user responds with the letter "Y", the system will ask for a database name and number; as follows:

NAME, NUMBER=?

The name can as long as sixteen characters and the number may be either one or two digits. The number should be separated from the name by a comma, and the whole followed by a carriage RETURN; for example:

TEST,1 (plus RETURN)

In response, the system will initialize the diskette. When this operation is complete, system command will be passed to the MAIN EDITING module.

2.3 System Commands

Most commands in the NETPAD system correspond to a single key on the PET keyboard, e.g., the "!" command, the "&" command, the "%" command, etc.

However, there are not enough special keys on the PET keyboard for all commands needed in NETPAD. The alphanumeric keys must therefore be used. To tell NETPAD that what you are typing is a command and not data, the "OFF/RVS" key must be pressed first.

Note that the "OFF/RVS" key is a dual function key. That is, this key may be shifted or unshifted. If the key is shifted using the "SHIFT" key the "OFF/RVS" key will transmit one meaning. If the "OFF/RVS" key is pressed in the unshifted mode the key will transmit another meaning. Accordingly, in this manual, the "OFF/RVS" key is referred to either as the "OFF/RVS" key or the "SHIFTED OFF/RVS" key. The shifted mode of other dual function keys will be similarly indicated, for example, the "SHIFTED CLR/HOME" key.

There are also two special keys on the PET keyboard. One marked "CRSR" with arrows pointing up and down and another marked "CRSR" with arrows pointing right and left. These keys will be referred to as the "CRSR UP/DOWN" key and the "CRSR RIGHT/LEFT" key. The shifted mode of these keys will be referred to in the normal way as the "SHIFTED CRSR UP/DOWN" key and the "SHIFTED CRSR RIGHT/LEFT" key.

3.0 SUBNETWORKS

3.1 Introduction

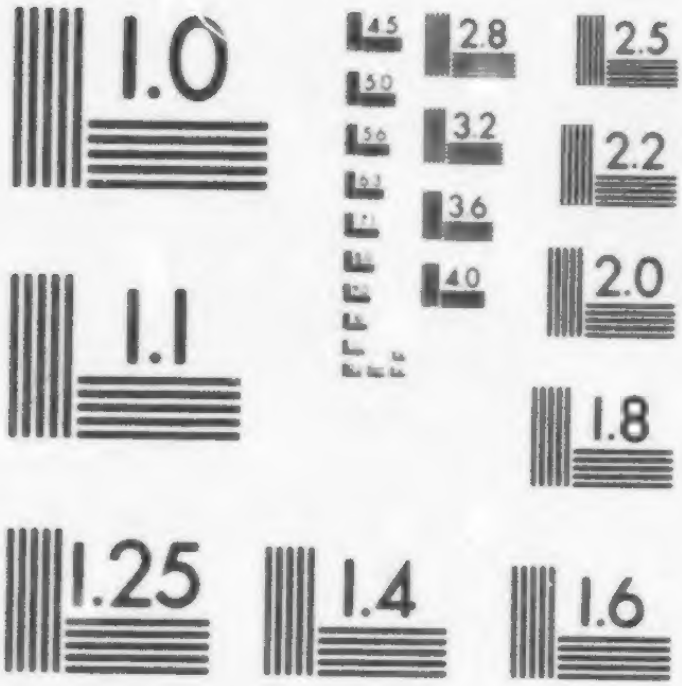
Subnetworks are the basic units of the NETPAD system. Each subnetwork takes the form of a set of concept nodes and a set of links between concept nodes. Because concept nodes are fundamental to subnetworks we will deal with them first. The use of links will be explained later.

3.2 Concept Nodes

In practice, subnetworks are most often used to describe single items covered by a database. For example, a subnetwork might represent a bibliographic item, say an article in a periodical, or a book.

For the sake of examples in the remainder of this manual, let us suppose that our subnetwork represents an article written in 1983 by a man called Robert Smith with the title "A feasibility study regarding the application of microcomputers in libraries". Such an item might be named usefully in several ways: by title, by author, by year, by subject, and so on.

In the NETPAD system these elements would be referred to as concept nodes. Thus, we would have a concept node identifying the author (in this case, Robert Smith), a concept node giving the year ("1983"), a concept node giving the title, and/or a number of concept nodes identifying the subject of the document.

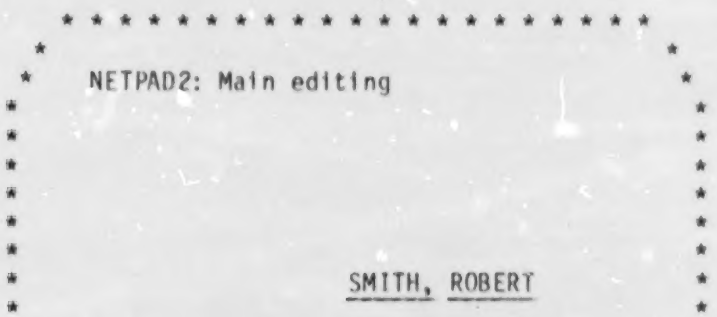


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

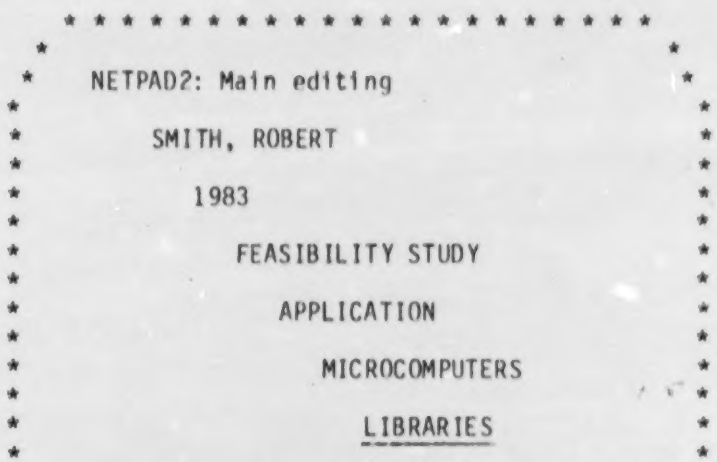
3.2.1 Inputting Concept Nodes

Concept nodes are input in the MAIN EDITING module. This module is NETPAD's default module and the first module a user will reach after starting the system.

Once within the MAIN EDITING module, a term or a string of characters beginning with a letter or a numeric may be typed in. If this entry is followed by a carriage RETURN, the term or string will mark the first node of the subnetwork and it will be displayed in the middle of the screen; for example:



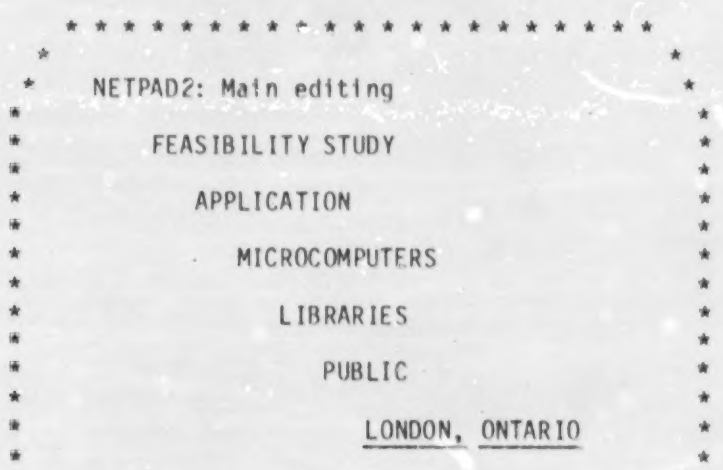
Other concept nodes may be appended using the same procedure. Note that all subsequent concept nodes will be displayed in a "staircase" fashion; for example:



Note also that the term or string last typed in always appears in reverse-field. That is, the characters representing the current node are displayed in white against a black background as opposed to being displayed in black against a white background.

The same effect however cannot be reproduced in the manual. For this reason plain underlining replaces reverse-field as the indicator of the current node in this manual.

At the input stage only six concept nodes may be simultaneously displayed on the screen. If a seventh concept node is input, the first will automatically scroll off the top of the screen and disappear out of view. The same will happen for any subsequent entries; as follows:



In order to view the first two concept nodes again the user must specifically adjust the "window" on the subnetwork being input. Four commands are available for this purpose, the "CRSR UP/DOWN" command, the "SHIFTED CRSR UP/DOWN" command, the "CRSR RIGHT/LEFT" command, and the "SHIFTED CRSR RIGHT/LEFT" commands.

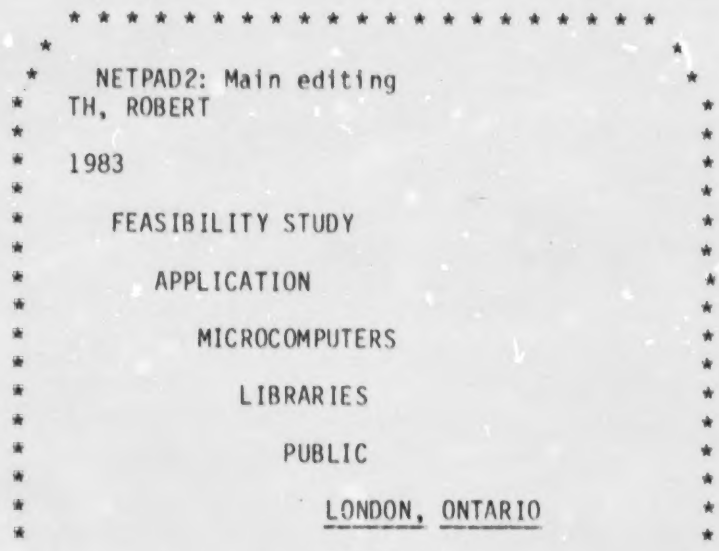
The "CRSR UP/DOWN" command moves the subnetwork "window" two lines up from its current position.

The "SHIFTED CRSR UP/DOWN" command moves the subnetwork "window" two lines down from its current position.

The "CRSR RIGHT/LEFT" command moves the subnetwork "window" 10 characters left from its current position.

The "SHIFTED CRSR RIGHT/LFFT" command moves the subnetwork "window" 10 characters right from its current position.

The third and fourth commands, the "CRSR RIGHT/LEFT" and the "SHIFTED CRSR RIGHT/LEFT" commands are needed in order to re-adjust the subnetwork "window". In the present example for instance, holding down the SHIFT key while striking the "CRSR UP/DOWN" key once will result in the following display:



Clearly we need the "CRSR RIGHT/LEFT" and the "SHIFTED CRSR RIGHT/LEFT" commands to properly adjust the "window". Notice also that at this stage eight concept nodes may now be displayed at one time.

3.2.2 Editing Concept Nodes

Concept nodes may be edited in the same module as they are input, i.e., the MAIN EDITING module.

There are two steps involved:

The first step consists in identifying the concept node we wish to edit. This is accomplished by moving the reverse-field to the wanted node. Two commands are available for this purpose, the ">" command and the "<" command.

> The ">" command moves the reverse-field one node down the "staircase" display.
>
> The "<" command moves the reverse-field one node up the "staircase" display.

Thus, in relation to the above example, if we wanted to edit the node represented by the term "FEASIBILITY STUDY" and the current node happened to be "SMITH, ROBERT" we would press the ">" key twice. Once to move the reverse-field down the "staircase", i.e., to "1983", and a second time to move the reverse-field from "1983" to the wanted node, i.e., "FEASIBILITY STUDY".

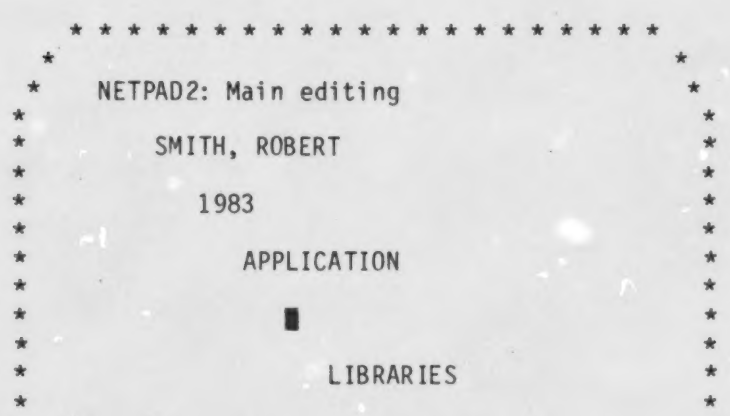
Note that if we wished we could move the reverse-field at will, that is up or down, within the "staircase" display.

The second step involved in editing a concept node is to replace the current term or phrase representing the concept node with a new or modified term or phrase. The command used for this purpose is the "!" command.

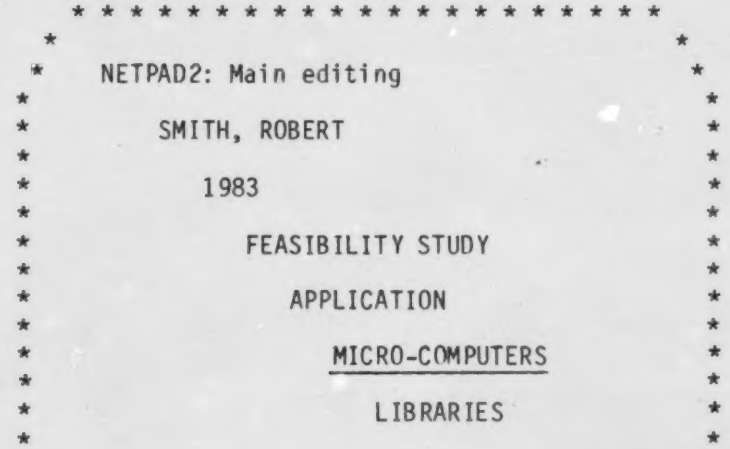
> The "!" command initiates the editing of the current concept node by (1) deleting the term or phrase representing the current node and (2) allowing the input of a new or modified term or phrase for that node.

Thus, if we wished to change the spelling of the term "MICROCOMPUTERS" to "MICRO-COMPUTERS" and the current node happened to be "FEASIBILITY STUDY" we would press the ">" key twice to move the reverse-field to the wanted node. Next we would press the "!" key in order to indicate our

intention of editing that concept node. The term "MICROCOMPUTERS" would be replaced by a flashing cursor (represented by a box shaped character in this manual) and the system would wait for the user to input the modified version of that concept node; as follows:



We would then type in the term "MICRO-COMPUTERS" and press the carriage RETURN key. This would result in the acceptance of the edited term and its proper display within the "staircase"; as follows:



3.2.3 Deleting Concept Nodes

Two steps are involved in deleting a concept node.

The first step consists in identifying the concept node we wish to delete. As in the editing procedure, this is accomplished by moving the reverse-field to the wanted node. The ">" command and the "<" command are available for this purpose.

> The ">" command moves the reverse-field one node down the "staircase" display.
>
> The "<" command moves the reverse-field one node up the "staircase" display.

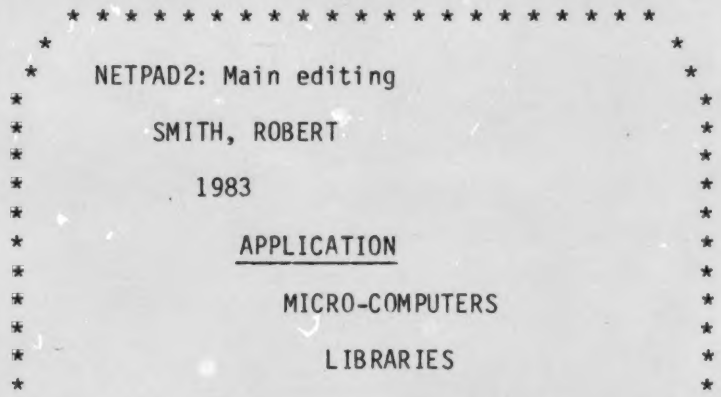
The second step involved in deleting a concept node is to indicate our intention to delete this node. This is accomplished by pressing the "#" key.

> The "#" command causes the deletion of the current concept node.

Thus, if we wished to delete the concept node represented by the term "FEASIBILITY STUDY" and the current node happened to be "SMITH, ROBERT" we would press the ">" key twice. Once to move the reverse-field to the concept node represented by the term "1983", and a second time to move the reverse-field to the wanted node. Next we would press the "#" key. In response, system command would be transferred to NETPAD's NODE DELETION module. The current concept node would be deleted and the following display would briefly appear on the screen:

```
* * * * * * * * * * * * * * * * * * * * * * *
*                                             *
*  NETPAD5: Node deletion                     *
*                                             *
*  DELETING                                   *
*  DONE                                       *
*                                             *
```

After a few seconds, system command would be returned to the MAIN EDITING module and the updated version of the subnetwork would be displayed on the screen in the standard "staircase" fashion; as follows:



3.3 Links

In the NETPAD system links refer to conceptual relations which exist between concept nodes. The type and kind of such relationships, their number and their expression usually differ among applications. For instance, relationships relevant to a manufacturing database may be relations of product, raw materials, and/or process. Relationships relevant to a bibliographic database may be relations of authorship, source, year, size, or subject.

3.3.1 Defining Linktypes

In the NETPAD system each type of relation is assigned a one-character mnemonic, a forward connective, a backward connective, and a set of weights. The whole is referred to as a linktype.

The one-character mnemonics uniquely identify each type of relation.

The connectives, one forward-reading for expressing the sense of a link of the given type from concept A to concept B and the other backward-reading from concept B to concept A, are used in the generation of index entries. Different indexes adopt different styles of connectives: some make extensive use of natural-language elements such as as prepositions (e.g., of, by, in); others prefer discreet punctuation to avoid distracting the searcher from the descriptors (such as the dash in Library of Congress subject headings).

The weights control the order in which the elements of the descriptions in the index display are cited.

These and other features of the NETPAD system will be discussed again later. For the moment let us note that in most cases the person setting up a NETPAD database will define his/her own linktypes and assign them the mnemonics, connectives, and weights most suited to his/her application. The system designer could use some of NETPAD's predefined

system linktypes, but these were specifically designed to be used in the ANALYSIS module and therefore may not be suitable for the current application.

For the sake of examples in the remainder of this manual, let us assume that our database is a bibliographic one. Let us also assume that the relations most relevant to that database are relations of authorship, publication date, and subject and that there are three relations relevant to the subject aspect of the documents within the database, relations of activity, relations of object, and relations of institution. Let us also assume that the most appropriate mnemonics, connectives, and weights are the following:

mnemonic	forward connective	weight	backward connective	weight
Y	. year:	10	.	5
S	. subject:	50	.	5
R	regarding	40	.	5
O	of	30	.	5
I	in	20	.	5

3.3.2 Inputting Linktypes

Linktypes are input in the REDEFINING module. The REDEFINING module is accessed by:

(1) pressing the "OFF/RVS" key, and

(2) following the prompt:

NETPAD0: Choosing
OPTION=?

typing in the command REDEFINE (plus RETURN).

At that point, the system will automatically display five standard system linktypes plus, if any are present, the user defined linktypes. (As said earlier, system linktypes are normally only used by the ANALYSIS module. They may be safely ignored for the moment.) The first time around however no user defined linktypes will be present and the linktype list will look as follows:

NETPAD1: Redefining

SUBHEADING: 0 CUTOFF: 0

#	MNEM.	FORWARD wt CONNECTIVE	BACKWARD wt CONNECTIVE
1	1	32	16
2	2	96	16
3	3	128	16
4	4	255	0
5	5	0	16
6			

Note that each linktype is numbered. This number is purely sequential and simply indicates the location of a particular linktype. Because of this numbering system we can refer to empty linktype positions. In the above linktype list for instance the first empty linktype position is linktype #6.

Note however that the cursor (represented by a box shaped character in this manual) is positioned next to linktype #1.

Each linktype must be entered separately. They are two steps involved:

The first step consists of positioning the cursor next to the first empty linktype position. Two commands are available for this purpose, the ">" command and the "<" command.

> The ">" command moves the cursor one position down the linktype list.
>
> The "<" command moves the cursor one position up the linktype list.

Thus, if we wanted to enter linktype "Y", and the cursor was positioned next to linktype #1 we would press the ">" key five times. Once to move the cursor down to the next linktype, i.e., linktype #2, and so on until the cursor would be positioned next to the first empty linktype, that is, linktype #6.

The second step involved in defining a linktype is to enter the mnemonic, the connectives, and the weights. This is accomplished by typing in the data in response to system prompts. The command used for this purpose is the "!" command.

> The "!" command (1) clears linktype #n, where n stands for the linktype indicated by the cursor, and (2) triggers a series of prompts which allow the input of linktype information; as follows:

```
 6
MNEMONIC=
FORWARD CONNECTIVE =
WEIGHT =
BACKWARD CONNECTIVE =
WEIGHT =
```

Note that except in the case of the first prompt the data entered must be followed by a carriage RETURN. This is necessary in order to indicate that (1) the entry is complete and that (2) the next prompt may be triggered. The last carriage RETURN triggers the display of the updated linktype list.

If we had entered all of the above defined linktypes via these prompts our linktype list would now look as follows:

```
NETPAD1: Redefining

 SUBHEADING: 0       CUTOFF: 0

 #    MNEM.   FORWARD        BACKWARD
               wt             wt
               CONNECTIVE     CONNECTIVE
 1    1       32             16
 2    2       96             16
 3    3       128            16
 4    4       255            0
 5    5       0              16
 6    Y       10             5
               . year:        .
 7    S       50             5
               . subject:     .
 8    R       40             5
               regarding      .
 9    0       30             5
               of             .
```

Note that the above display does not show all of the linktypes we are supposed to have defined. This is because the PET2001 microcomputer screen is limited to 25 rows of data. In order to see the remainder of the linktype list we must to specifically ask for it. The command used for this purpose is the "CRSR UP/DOWN" command.

> The "CRSR UP/DOWN" command displays the next screenful of linktypes.

Thus, in order to see the remainder of the linktype

list we should press the "CRSR UP/DOWN" key. In response, the NETPAD system would display the next screenful of linktypes; as follows:

```
  #   MNEM.   FORWARD        BACKWARD
                wt             wt
              CONNECTIVE     CONNECTIVE
  10    I       20             5
                in             .
 ■11
```

If at this point we wanted to look at the first screenful we would have to press the "CLR/HOME" key.

> The "CLR/HOME" command displays the first screenful of linktypes.

If we wanted to exit from the REDEFINING module, we could press any key which does not constitute a command to the REDEFINING module (e.g., the space bar). As a result, system command would be transferred back to the MAIN EDITING module.

3.3.3 Editing Linktypes

If for one reason or other we had made a mistake or, if we wished to experiment with different connectives or weights we may want to edit existing linktypes.

This can also be done in the REDEFINING module. The procedure is exactly the same as the one involved in defining a new linktype. The only difference is that in this case the cursor which is normally used to indicate an empty linktype position should be used to indicate the position of the linktype we wish to edit.

3.3.4 Assigning Links between Concept Nodes

There are four basic steps involved in assigning links between concept nodes:

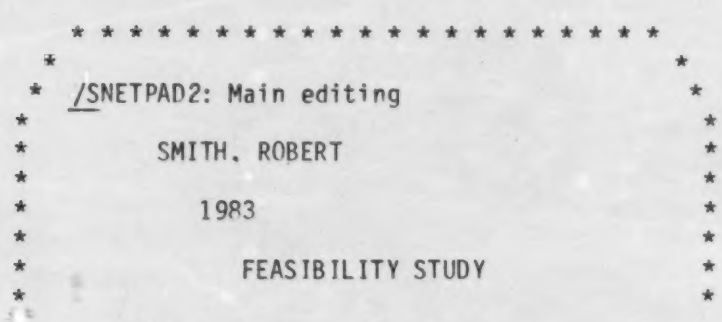
The first step consists in identifying the source node, that is, the node from which a link needs to be established. This is accomplished by moving the reverse-field from the current reverse-field node to the wanted node. Two special commands are available for this purpose, the ">" command and the "<" command.

> The ">" command moves the reverse-field one node down the "staircase" display.
>
> The "<" command moves the reverse-field one node up the "staircase" display.

Thus, if we wanted to assign a link from the node represented by the term "FEASIBILITY STUDY" and the current reverse-field node happened to be "SMITH, ROBERT" we would press the ">" key twice. Once to move the reverse-field to the next node down the "staircase", i.e., "1983", and a second time to move the reverse-field from "1983" to the wanted node, i.e., "FEASIBILITY STUDY".

The second step involved in assigning a link is to declare the linktype we wish to use. This is accomplished by typing in one of the previously defined one-character linktype mnemonics . The command used for this purpose is the "/" command. (There is an other suitable command, but it is more difficult to learn and will be dealt with later.)

> The "/" command initiates a link from the currently reverse-field node to another node. Its syntax is "/n", where n stands for a one-character mnemonic referring to a linktype. A helpful feature of this command is that when both the "/" key and the one-character mnemonic have been pressed both the command and the linktype mnemonic appear in a flashing mode in the upper left corner of the screen. This reminds the user of the linktype chosen; for example:



The third step involved in assigning a link is to identify the destination node, that is, the node to which the link needs to be established. This is accomplished by moving the reverse-field to the destination node. The same commands used in step one, the ">" and "<" commands. may be used for this purpose.

The fourth step necessary to assign a link between concept nodes is to indicate that a link is complete. This is accomplished by pressing the "↑" key.

> The "↑" command indicates that a link or unlinking operation has been completed and triggers the display of the updated subnetwork on the screen.

In summary, if we wished to assign a link of type "Y" between the nodes "SMITH, ROBERT" and "1983" and the current reverse-field node happened to be "LIBRARIES" we would first press the "<" key five times to move the reverse-field to the source node. Second we would press the "/" key and type in the linktype mnemonic "Y" to indicate that we wished to establish a link of that type. Third we would press the ">" key in order to move the reverse-field from the source node to the destination node. Fourth, and finally, we would press the "↑" key to indicate that the link is complete.

At that time, a link between these nodes would be established and the subnetwork would look as follows:

```
NETPAD2: Main editing

    SMITH, ROBERT
     /Y
       1983

          FEASIBILITY STUDY

             APPLICATION

                MICROCOMPUTERS

                   LIBRARIES
```

If we proceeded to assign all the relevant links to the subnetwork, that is, links of type "S", "R", "O", and "I", the resulting subnetwork would look as follows:

```
NETPAD2: Main editing

    SMITH, ROBERT
    I/Y
    I  1983
     /S
       ---FEASIBILITY STUDY
          /R
            APPLICATION
             /O
               MICROCOMPUTERS
                /I
                  LIBRARIES
```

3.3.5 Unlinking Concept Nodes

If for one reason or other we had made a mistake, or if we wished to experiment, we may want to unlink presently linked concept nodes.

The procedure is exactly the same as the one involved in assigning links. The only difference is that instead of indicating the linktype we wish to assign we must indicate that we wish to unlink. The command used for this purpose is the "<" command.

> The "<" command initiates unlinking from the current reverse-field node to some other node.

Thus, if we wanted to delete the link between concept node "MICROCOMPUTERS" and concept node "LIBRARIES" and the current reverse-field node was "SMITH, ROBERT" we would first press the ">" key four times to move the reverse-field to the source node. Second, we would press the "<" key to indicate that we wish to initiate the unlinking of that node from another node. Third, we would press the ">" key once in order to move the reverse-field to the destination node. Fourth, and finally, we would press the "↑" key to complete the unlinking operation.

At that time, the link which had been present between these two nodes would be deleted and the subnetwork would look as follows:

```
NETPAD2: Main editing

        SMITH, ROBERT
        I/Y
        I  1983
         /S
           ---FEASIBILITY STUDY
                /R
                  APPLICATION
                   /O
                     MICROCOMPUTERS

                         LIBRARIES
```

3.4 Appending Concept Nodes And Links

Concept nodes may be appended to the current subnetwork at any stage during its construction. If a new concept node is appended however, that node will appear at the bottom of the "staircase" display. Existing nodes and/or links are never moved in order to make room for the new node.

Note that, if at this stage a link needs to be established it is not necessary to laboriously move the reverse-field to the source node. If we wish to initiate a link from an existing node to the newly appended node without moving the reverse-field we can use the "\" command.

> The "\" command initiates a link to the currently reverse-field node from another node. (Its syntax is the same as the "/" command)

If, on the other hand, we wished to initiate a link to an already existing node from the newly appended node without moving the reverse-field, we would use the standard and more often used "/" command.

> The "/" command, as previously explained, initiates a link from the currently reverse-field node to another node.

Since in such a case the order of the concept nodes on the screen would be reversed (i.e., the source node follows instead of precedes the destination node) the slash which normally precedes the linktype mnemonic in the "staircase" display would follow the linktype mnemonic, thereby indicating that the direction of the link is in reverse order.

For example, if within the sample subnetwork displayed above, we wished to add a concept node represented by the term "MINICOMPUTERS" and link it in the same way as the concept node represented by the term "MICROCOMPUTERS" is linked to the concept node represented by the term "APPLICATION", we would first type the term in and then use the "\" command to establish the wanted link. We would use the "\" command instead of the "/" command because, in this case, we want to establish a link to and not from the current node. The resulting subnetwork would look as follows:

```
NETPAD2: Main editing

    SMITH, ROBERT
    I/Y
    I  1983
     /S
       ---FEASIBILITY STUDY
          /R
            APPLICATION
            I/O
            I  MICROCOMPUTERS
            I   /I
            I     LIBRARIES
             /O
               ------MINICOMPUTERS
```

On the other hand, if we wished to add a concept node represented by the term "COST ANALYSIS" and link it to the concept node represented by the term "FEASIBILITY STUDY" we would use the "/" command. We would use the "/" command because, in this case, we want to establish a link from the current node. The resulting subnetwork will look as follows:

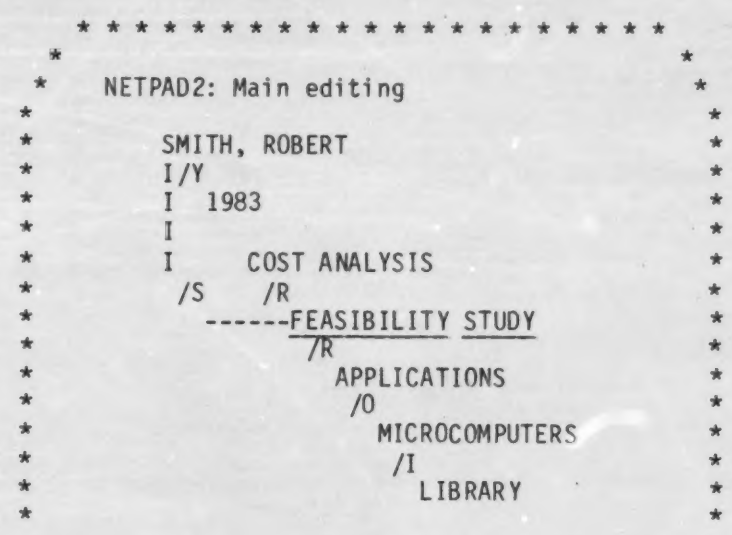
```
NETPAD2: Main editing

    SMITH, ROBERT
    I/Y
    I  1983
     /S
       ---FEASIBILITY STUDY
          I/R
          I  APPLICATION
          I   /O
          I     MICROCOMPUTERS
          I      /I
          I        LIBRARIES
           R/
             ---------COST ANALYSIS
```

3.5 Sorting Subnetworks

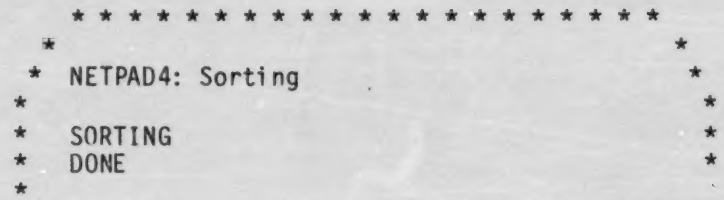
It seems reasonable to assume that the content of a subnetwork will more readily be grasped and validated (for example, by comparison with similar subnetworks) if the elements were to be arranged on the screen according to a recognized set of principles. Accordingly, NETPAD permits the user to restructure displays so that the nodes in the "staircase" appear in a standard citation order. The module used for this purpose is the SORTING module and it is accessed by pressing the "'" key.

> The "'" command transfers system command to the SORTING module which orders the concept nodes of the current subnetwork first according to the direction of the linktypes, then according to the precedence order among linktypes (as determined by the weights), and finally, according to the alphabetical order of the terms marking the concept nodes.

For example, the sample subnetwork displayed in the previous section would appear, in sorted order, as follows:



Note that the sorting operation will take a few seconds and that during this time the following display will briefly appear on the screen:



More important however, is to note how the sort module removes the inverted links. The only exception to this rule will occur when there is at least one circuit in the subnetwork, resulting in incompatible demands among links within a set of concepts. Sorting also usually minimizes the number of crossovers among the lines indicating links.

4.0 DISPLAYING SUBNETWORKS IN TABLE FORM

If desired, subnetworks may be displayed in the form of two tables, one a table of nodes, and the other a table of triads representing links, each triad consisting of a node number, a linktype number, and a second node number. The command used for this purpose is the "?" command.

The "?" command triggers the display of the current subnetwork in table form; as follows:

```
NETPAD6: Table display

 #    TERM
 1    SMITH, ROBERT
 2    1983
 3    FEASIBILITY STUDY
 4    APPLICATION
 5    MICROCOMPUTERS
 6    LIBRARIES
 #    LINKTYPES  #
 1      10       2
 1      14       3
 3      11       4
 4      12       5
 5      13       6
```

This form of display closely echoes the way the data is stored in computer memory, and is most likely to be used by the programmer for diagnostic purposes. We have included it here however for your information prior to our discussing the procedure involved in storing subnetworks.

Pressing any key will transfer system command back to the MAIN EDITING module.

5.0 STORING SUBNETWORKS

Once a subnetwork has been constructed the normal procedure is to store it within the NETPAD database for later retrieval. The command used for this purpose is the ")" command.

> The ")" command appends the current subnetwork to the current NETPAD database.

This procedure may take a while depending on how many subnetworks already exist within the database. As the ")" command performs its functions, the track and sector numbers of the diskette blocks which have been modified will one after the other be displayed on the screen; as follows:

```
NETPADS: Storing
 2        0
 1        4
 2        3
 1        4
 1        4
 1        4
 1        4
 1        4
 1        4
 1        4
```

Following the successful completion of this procedure the system will (1) transfer system command back to the MAIN EDITING module and (2) re-display the subnetwork.

6.0 CLEARING THE WORKSPACE

As noted above, the ")" command does not automatically clear or erase the current subnetwork from the NETPAD workspace. On the contary, it re-displays the subnetwork on the screen. Therefore, if at that point we wished to construct a new subnetwork we would first have to clear or erase the current subnetwork from the workspace. The command used for this purpose is the "SHIFTED CLR/HOME" command.

> The "SHIFTED CLR/HOME" command clears the current subnetwork from the workspace.

That is, the current subnetwork is erased from computer memory and the screen .

7.0 RETRIEVING SUBNETWORKS

Subnetworks may be retrieved via two different modules:

(1) via the QUERYING module, or

(2) via the BROWSING module.

The simplest method of retrieving subnetworks is via the QUERYING module. This module also automatically displays the retrieved subnetworks in permuted index entry form. The browsing module on the other hand offers Boolean searching capability.

7.1 Retrieving Subnetworks Via The QUERYING Module

The QUERYING module is accessed by:

(1) pressing the "&" key, and

(2) following the prompt:

```
NETPADQ:  Querying
KEY=?
```

typing in a search key, which may be truncated (plus RETURN). Note that the search key should be enclosed in double quotes if it contains a comma or a colon (e.g., "SMITH, ROBERT").

In response, the system will retrieve all the subnetworks within the database which have a term beginning with the specified key. For example, in response to the search key "SMITH", the system might retrieve the following subnetworks:

```
NETPADR: Retrieving

 SMITH, JOHN . subject: SURVEY
     regarding CLASSIFICATION of
     SLAVIC MATERIALS in ACADEMIC
     LIBRARIES . year: 1972
 SMITH, KEN . subject: SYSTEM
     ANALYSIS regarding MANAGEMENT of
     SERIALS in SLIS . year: 1975
 SMITH, ROBERT . subject:
     FEASIBILITY STUDY regarding
     APPLICATION of MICROCOMPUTERS in
     LIBRARIES . year: 1983
 SMITH, WILLIE . subject:
     STATISTICAL ANALYSIS regarding
     CIRCULATION of AV MATERIALS in
     SCHOOL LIBRARIES . year: 1981
```

Note that all the subnetworks are displayed in permuted form with each term linked with our previously defined connectives. As we will see later, the format of this display is partly dependent on user defined parameters and may therefore be changed. Thus, for the moment let us ignore the format of the entries and simply note that a flashing cursor appears next to the first entry on the screen.

Sometimes not all the permutations generated in response to a search key can be displayed on a single screen. If the search key is a very common term (e.g., the personal name SMITH) or a truncated term (e.g., the letter A) ten or more screenfuls of permutations could easily be generated. These will not automatically be displayed on the screen. They must be called up manually. The command used for this purpose is the "CRSR UP/DOWN" command.

> The "CRSR UP/DOWN" command displays the next screenful of permutations.

If at any time while displaying these additional screens we wished to return to the first screenful we can press the "CLR/HOME" key.

> The "CLR/HOME" command displays the first screenful of permutations.

Once the appropriate screenful of permutations is selected two additional steps are necessary to retrieve a subnetwork.

The first step consists of positioning the cursor next to the subnetwork we wish to retrieve. The commands used for this purpose are, the ">" command and the "<" command.

> The ">" command moves the cursor one entry down the list of permutations.
>
> The "<" command moves the cursor one entry up the list of permutations.

Thus, if we wanted to retrieve the subnetwork with the name "SMITH, ROBERT" and the cursor was positioned next to the first entry we would press the ">" key twice. Once to move the cursor down to the next entry, i.e., "SMITH, KEN", and a second time in order to move the cursor from "SMITH, KEN" to the wanted entry, i.e., "SMITH, ROBERT".

The second step involved in retrieving a subnetwork via the QUERYING module is to:

> (1) press the "↑" key, and
>
> (2) following the prompt:
>
> DELETE?
>
> type in the letter "N".

In response, the NETPAD system will (1) retrieve the selected subnetwork from the NETPAD database, (2) display the selected subnetwork on the screen in the standard "staircase" fashion, and (3) pass system command to the MAIN EDITING module.

Note that, if the user does not want to retrieve a subnetwork and wishes to return to the MAIN EDITING module, the "SHIFT ↑" key can be used to exit from the QUERYING module.

The "SHIFTED ↑" command transfers system command back to the MAIN EDITING module.

7.2 Retrieving Subnetworks Via The BROWSING Module

The BROWSING module retrieves subnetworks by a method known as inverted-index searching. It is not unlike the way a person searches for information in a book. Most books have an index. Suppose, for example, that the book is on Canadian history and the topic is the Railways. The book is opened to the index at the back and the index is scanned until the alphabetical listing starting with R is found. The R list is scanned for Ra, then Rai, then Rail, etc., until the term Railways is found. The entry found will probably look like:

Railways, 10, 48, 76, 109, 116, 238.

The list of numbers following the term indicates which pages of the book contain information on Railways.

A typical search within the BROWSING module proceeds as follows. The user enters a search key. The system scans the inverted-index file for a term equal to or beginning with the same characters as the search term. If such a term is not present in the index file the execution of that search is aborted. If a term is present in the index file, the number of the subnetworks containing the term is displayed on the screen, e.g., 6 ITEM(S). The diskette addresses of relevant subnetworks are saved as a set. This set may then be combined with other sets via more complex search formulations or, alternatively, the diskette addresses of the relevant subnetworks may be displayed on the screen using the "SHIFT G" command.

The BROWSING module is accessed by:

(1) pressing the CLR/RVS key, and

(2) following the prompt:

```
NETPADO:  Choosing
OPTION=?
```

typing in the command BROWSE (plus RETURN).

At that point, the system will automatically display a brief summary of the commands available within this module;

as follows:

```
* * * * * * * * * * * * * * * * * * * * * * * *
*                                               *
*  NETPADB: Browsing                            *
*                                               *
*  !FIND   ↑STORE   ↑EXIT   >NEXT               *
*  +OR  *AND  -BUT NOT  #SET                    *
*  gGET                                         *
*                                               *
```

There are nine commands including three Boolean operators. The most commonly used command is the "!" command.

> The "!" command retrieves the diskette addresses of the subnetworks which have a concept node with a term corresponding to the specified search key.

The command is activated by:

> (1) pressing the "!" key, and
>
> (2) following the prompt:
>
> !?
>
> typing in a search key (plus RETURN). Note that the search key should be enclosed in double quotes if it contains a comma or a colon (e.g., "SMITH, ROBERT").

In response, the system will (1) re-display the search key and (2) indicate the number of subnetworks which have a concept node with a term corresponding to the search key; as follows:

```
* * * * * * * * * * * * * * * * * * * * * * * *
*                                               *
*  NETPADB: Browsing                            *
*                                               *
*  !FIND   ↑STORE   ↑EXIT   >NEXT               *
*  +OR  *AND  -BUT NOT  #SET                    *
*  gGET                                         *
*  !? MICROCOMPUTERS                            *
*  MICROCOMPUTERS       4 ITEM(S)               *
*                                               *
```

If the search fails, that is, if the search key is not present in the inverted-index file, the execution of the command will stop. No message will be displayed and the module will only be re-activated if the user enters a new command.

The subnetwork addresses retrieved are now said to be the current set and, as said earlier, the current set may be stored for later use. The command used for this purpose is the "↑" command.

> The "↑ " command stores the current set of subnetwork addresses as the next set in a set list whose sets are identified by a single letter of the alphabet starting with the letter "A".

Thus, if we wished to save the results of our latest search prior to initiating another search, we would press the "↑" key. In response, the system would (1) store the result of our latest search, (2) label it using a one-character letter code, and (3) display the letter code on the screen; as follows:

```
* * * * * * * * * * * * * * * * * * * * * * *
*                                             *
*  NETPADB: Browsing                          *
*                                             *
*  !FIND   ↑STORE   ↑EXIT   >NEXT             *
*  +OR  *AND  -BUT NOT  #SET                  *
*  gGET                                       *
*  !? MICROCOMPUTERS                          *
*  MICROCOMPUTERS       4 ITEM(S)             *
*  ↑SET  A                                    *
*                                             *
```

We can now safely execute another search without fear of loosing the result of our previous search. For example, we could now search for all the subnetworks which have a concept node marked by the term "1983"; for example:

```
NETPADB: Browsing

!FIND   ↑STORE   ↑EXIT   >NEXT
+OR  *AND  -BUT NOT  #SET
gGET
!? MICROCOMPUTERS
MICROCOMPUTERS        4 ITEM(S)
↑SET   A
!? 1983
1983       2 ITEM(S)
```

We now have two sets of subnetwork addresses: set A containing the addresses of all the subnetworks which have a concept node marked by the term "MICROCOMPUTERS", and the current set which contains the addresses of all the subnetworks which have a concept node marked by the term "1983". If at this stage we wished to find out which subnetworks have concept nodes with both search terms, or which subnetworks have concept nodes with one term but not the other, or which subnetworks have concept nodes with one search term but not the other, we could use the Boolean operators: AND, OR, BUT NOT.

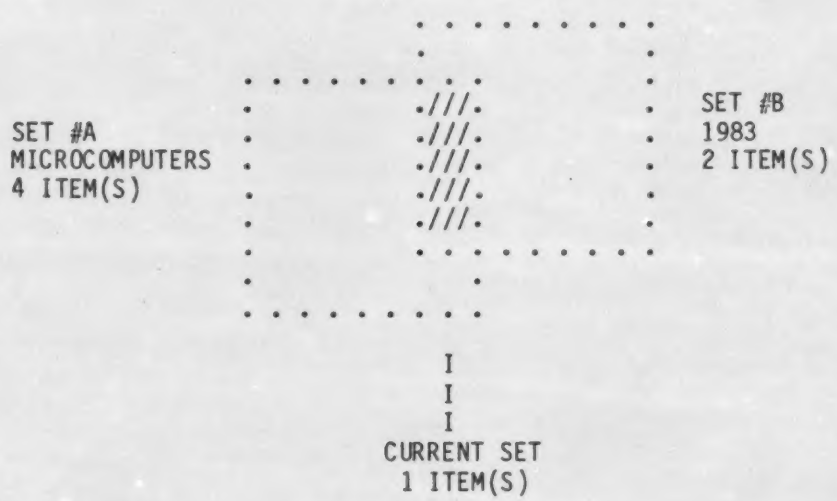
Combining sets by means of Boolean operators makes it possible to define searches such as "MICROCOMPUTERS and 1983" so as to retrieve only the subnetworks which have concept nodes marked by both search keys.

For brevity, the Boolean operators are represented by special characters: * for AND, + for OR, and - for BUT NOT. The syntax of these operators is *n, +n, -n, where n stands for a letter representing a previously stored set.

In the following example, the current set (i.e., the set of subnetworks which have a concept node marked by the term "1983") is intersected with the stored set represented by the letter code A (i.e., the set of subnetworks which have a concept node marked by the term "MICROCOMPUTERS):

```
NETPADB: Browsing

!FIND   ↑STORE   ↑EXIT   >NEXT
+OR  *AND  -BUT NOT  #SET
gGET
!? MICROCOMPUTERS
MICROCOMPUTERS       4 ITEM(S)
↑SET  A
!? 1983
1983       2 ITEM(S)
*A
1983 AND (MICROCOMPUTERS)      1 ITEM(S)
```

As a result, we can be sure that the subnetwork retrieved has a node marked by the term "MICROCOMPUTERS" and a node marked by the term "1983". The same search expression may be illustrated by the following diagram:



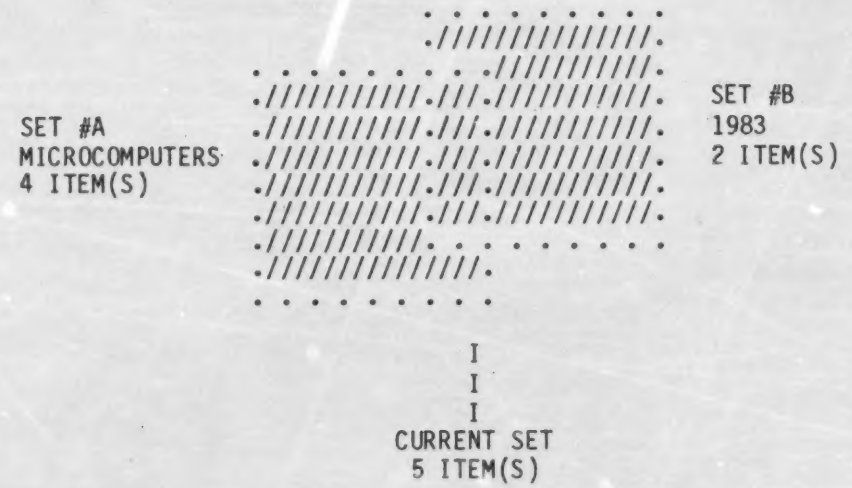
Note that the use of the AND operator almost always gives output smaller than each of the parts.

The OR operator is represented by the "+" character and it requests the presence of either of the search terms as concept nodes in the retrieved subnetworks; for example:

```
NETPADB: Browsing

!FIND   ↑STORE   ↑EXIT   >NEXT
+OR  *AND  -BUT NOT  #SET
gGET
!? MICROCOMPUTERS
MICROCOMPUTERS        4 ITEM(S)
↑SET  A
!? 1983
1983        2 ITEM(S)
+A
1983 OR (MICROCOMPUTERS)        5 ITEM(S)
```

The subnetworks retrieved by this operation include either the term "MICROCOMPUTERS" OR the term "1983" as concept nodes. In the event that both of these terms are present in the same subnetwork, that subnetwork will only be counted once.

The following diagram indicates the execution of the Boolean OR operator:



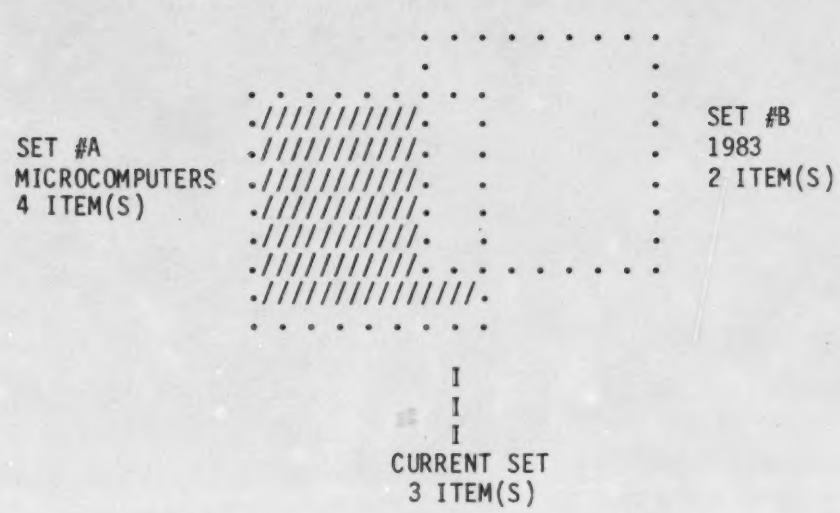
Note that the number of subnetworks retrieved is less than the sum of the two parts. This is because one of the subnetworks contains both of term "MICROCOMPUTERS" and the term "1983" and is counted only once. In general, use of the OR operator results in a larger output of subnetworks.

The BUT NOT operator removes some elements from the current set. It will retrieve subnetworks with one search term, BUT NOT if the other term is present; for example:

```
NETPADB: Browsing

!FIND   ↑STORE   ↑EXIT   >NEXT
+OR  *AND  -BUT NOT  #SET
gGET
!? MICROCOMPUTERS
MICROCOMPUTERS        4 ITEM(S)
↑SET   A
!? 1983
1983        2 ITEM(S)
-A
1983 BUT NOT (MICROCOMPUTERS)   1 ITEM(S)
```

This search indicates that of the 2 subnetworks that include the term "1983" there is 1 that does not have the term "MICROCOMPUTERS".

The diagram of the search with the BUT NOT operator is as follows:



If at this point we wished to recall a previously stored set of subnetwork addresses in order to attempt a different search strategy we would use the "#" command.

> The "#" command transfers a previously stored set of subnetwork addresses into current memory. That is, it makes a previously stored set the current set. The syntax of the command is "#n" where n stands for the letter code of a previously stored set.

The following example for instance shows the recall of the set represented by the letter A, i.e., the set of subnetworks which have a concept node marked by the term "MICROCOMPUTERS":

```
NETPADB: Browsing

!FIND   ↑STORE   ↑EXIT   >NEXT
+OR  *AND  -BUT NOT  #SET
gGET
!? MICROCOMPUTERS
MICROCOMPUTERS        4 ITEM(S)
↑SET   A
!? 1983
1983       2 ITEM(S)
-A
1983 BUT NOT (MICROCOMPUTERS)   1 ITEM(S)
#A
MICROCOMPUTERS        4 ITEM(S)
```

The current set now contains the diskette addresses of 4 subnetworks, each having a concept node marked by the term "MICROCOMPUTERS".

Another useful command is the ">" command.

> The ">" command transfers into current memory the set of subnetwork addresses attached to the next search term in the inverted-index file. That is, it automatically retrieves the diskette addresses of the subnetworks which have a concept node with a term corresponding to the next term in the inverted-index file.

Thus, in keeping with the above example, if we pressed the ">" key the current set would automatically be replaced by the set of the subnetwork addresses attached to the next key in the inverted-index file. In this case, it would be the set corresponding to the term "SCHOOL LIBRARIES", and there would be 2 such subnetworks; for example:

```
NETPADB: Browsing

!FIND   ↑STORE   ↑EXIT   >NEXT
+OR  *AND  -BUT NOT  #SET
gGET
!? MICROCOMPUTERS
MICROCOMPUTERS        4 ITEM(S)
↑SET  #A
!? 1983
1983       2 ITEM(S)
-A
1983 BUT NOT (MICROCOMPUTERS)   1 ITEM(S)
#A
MICROCOMPUTERS        4 ITEM(S)
>SCHOOL LIBRARIES       2 ITEM(S)
```

If, at that stage, we had not found the desired subnetwork we could elect to clear the screen and erase the set list in order to begin a new search session. The command used for this purpose is the "SHIFTED CLR/HOME" command.

The "SHIFTED CLR/HOME" command clears the screen, erases the set list and the current set, and re-displays the module command summary at the top of the screen.

Alternatively, we could elect to leave the module entirely. The command used for this purpose is the "SHIFTED ↑" command.

The "SHIFTED ↑" command clears the screen and returns system command to the MAIN EDITING module.

On the other hand, if we had found the desired subnetwork we could elect to retrieve it from the database. Three steps are involved.

The first step consists in calling up the subnetwork addresses listed in the current set. The command used for this purpose is the "SHIFTED G" command.

The "SHIFTED G" command transfers system execution to the GETTING module which in turn displays the diskette track and sector numbers of the subnetworks listed in the current set.

Such a display might look as follows:

```
  * * * * * * * * * * * * * * * * * * * * * *
 *                                           *
*  NETPADG: Getting                          *
*                                             *
*   ▮2  0                                     *
*    2  12                                    *
*    2  13                                    *
*    2  15                                    *
*                                             *
```

Note that a flashing cursor (represented by a box shaped character in this manual) appears next to the first diskette address.

The second step required to retrieve a specific subnetwork is to position the cursor next to the subnetwork one wishes to retrieve. The commands used for this purpose are the ">" command and the "<" command.

> The ">" command moves the cursor one entry down the list of subnetwork addresses.
>
> The "<" command moves the cursor one entry up the list of subnetwork addresses.

Thus, if we wished to retrieve the last subnetwork in the list and the cursor was positioned next to the address of the first subnetwork we would press the ">" key three times. Once to move the cursor to track 2 sector 12, a second time to move the cursor to track 2 sector 13, and a third time to move the cursor to the address of the wanted subnetwork, i.e., to the subnetwork with the address of track 2 sector 15.

The third step involved in retrieving a subnetwork via the BROWSING module is the same as the second step involved in retrieving a subnetwork via the QUERYING module. That is, we must:

> (1) press the "↑" key, and
>
> (2) following the prompt:
>
> DELETE?
>
> type in the letter "N".

In response, the NETPAD system will (1) retrieve the selected subnetwork from the NETPAD database, (2) display the selected subnetwork on the screen in the standard "staircase" fashion, and (3) pass system command to the MAIN EDITING module.

8.0 DELETING SUBNETWORKS

The procedure involved in deleting a subnetwork is exactly the same as the one involved in retrieving a subnetwork in either the QUERYING or the BROWSING modules. The only difference is that instead of typing in the letter "N" following the prompt "DELETE?" we should type in the letter "Y".

In response, the NETPAD system will transfer system command to the DELETING module. This module will automatically delete the selected subnetwork from the NETPAD database and indicate the successful completion of the operation by displaying on the screen the track and sector numbers of the diskette block freed; for example:

```
  * * * * * * * * * * * * * * * * * * * * * *
 *                                           *
*  NETPADD: DELETING                         *
*                                             *
*  TRACK 2   SECTOR 0   FREED                 *
*                                             *
*                                             *
```

Following this operation, system command will be returned to the MAIN EDITING module.

9.0 GENERATING PERMUTED INDEX ENTRIES

Permuted index entries for the current subnetwork may be generated by pressing the "%" key.

> The "%" command generates permuted index entries for the current subnetwork.

In response, the system will (1) transfer system command to the PERMUTING module and (2) proceed with the generation of permuted index entries. Because the permutations must be algorithmically built using a variety of parameters this process may take a while. In time however, appropriate permutations for the current subnetwork will be displayed on the screen; as in the following example:

```
NETPADP: Permuting

1983 . SMITH, ROBERT . subject:
    FEASIBILITY STUDY regarding
    APPLICATION of MICROCOMPUTERS in
    LIBRARIES
APPLICATION of MICROCOMPUTERS in
    LIBRARIES . FEASIBILITY STUDY .
    SMITH, ROBERT . year: 1983
FEASIBILITY STUDY regarding
    APPLICATION of MICROCOMPUTERS in
    LIBRARIES . SMITH, ROBERT . year:
    1983
LIBRARIES . MICROCOMPUTERS .
    APPLICATION . FEASIBILITY STUDY .
    SMITH, ROBERT . year: 1983
MICROCOMPUTERS in LIBRARIES .
    APPLICATION . FEASIBILITY STUDY .
    SMITH, ROBERT . YEAR: 1983
SMITH, ROBERT . subject:
    FEASIBILITY STUDY regarding
    APPLICATION of MICROCOMPUTERS in
    LIBRARIES . year: 1983
```

Sometimes not all the permutations generated from a subnetwork may be displayed on a single screen. If a subnetwork is very long or complex for instance, one or two additional screenfuls may be generated. These will not automatically be displayed. They must be called manually. The command used for this purpose is the "CRSR UP/DOWN" command.

> The "CRSR UP/DOWN" command displays the next screenful of permutations.

If at that point we wished to return to the first screenful of permutation we must press the "CLR/HOME" key.

> The "CLR/HOME" command displays the first screenful of permutations.

10.0 CUSTOMIZING PERMUTED INDEX DISPLAYS

The NETPAD system offers a variety of options which allow the user to customize permuted index displays. These include:

(1) the specification of different citation orders;

(2) the exclusion of selected item descriptions from index display;

(3) the control over boundaries between headings and subheadings; and

(4) the rewording of connective expressions.

10.1 Specification Of Different Citation Orders

To control the order in which elements of descriptions are cited in the index display, one need only reset the weights assigned to the appropriate connectives.

As we know, each linktype is assigned two connectives, usually prepositions or punctuation, one (forward-reading) for expressing the sense of a link of the given type from concept A to concept B and the other (backward-reading) from concept B to concept A; and two weights, one for each connective.

The weights indicate the strength of the bond the linktype represents from the first term in the link to the second and from the second to the first respectively. Remaining elements of the description joined to the first part of an index entry by the highest weights will be cited first.

Thus, in the following example:

```
NETPAD2: Main editing

    APPLICATIONS
    I/O
    I  MICROCOMPUTERS
     /I
       ---INDEXING
```

If the linktype represented by the mnemonic "O" is given a high weight relative to the linktype represented by the mnemonic "I", the index entry generated via the search key "APPLICATION" would appear as:

```
NETPADR: Retrieving

 APPLICATIONS of MICROCOMPUTERS in
     INDEXING
```

Whereas, if the relative higher weight were assigned to the linktype represented by the mnemonic "I", the entry would appear as:

```
NETPADR: Retrieving

 APPLICATIONS in INDEXING of
     MICROCOMPUTERS
```

Note that the precedence order represented by the weights applies throughout the index entry, at all levels of relationship to the access point.

```
NETPAD2: Main editing

OPINIONS
I/O
I  INDEXERS
I  I/I
I  I  CANADA
I   /Y
I     ---1980'S
 /R
   ---------ABSTRACTORS
            I/I
            I  GREAT BRITAIN
             /Y
               ---1890'S
```

In the above example for instance, if the linktype represented by the mnemonic "Y" is given a high weight relative to the linktype represented by the mnemonic "I", the precedence order represented by these weights would be applied throughout the index entry, as follows:

```
NETPADR: Retrieving

 OPINIONS of INDEXERS . year(s):
     1980'S in CANADA regarding
     ABSTRACTORS . year(s): 1890'S in
     GREAT BRITAIN
```

For procedure on how to reset weight values, see Section 10.5.

10.2

Exclusion of Selected Item Descriptions

Users will not always wish to see complete descriptions of indexed items under the access points looked up. The system therefore permits the specification of a cutoff threshold value, above which a linktype weight must fall in order for a linked term to be included in the index entry.

For example, this cutoff threshold might be set below the forward weight of linktype "O" but above the forward weight of linktype "I" (e.g., to 25), resulting in the index entry:

```
   * * * * * * * * * * * * * * * * * * * * * * *
  *                                               *
 *  NETPADR: Retrieving                           *
*                                                  *
*    APPLICATIONS of MICROCOMPUTERS                *
*                                                  *
```

in place of the fuller form:

```
   * * * * * * * * * * * * * * * * * * * * * * * *
  *                                               *
 *  NETPADR: Retrieving                           *
*                                                  *
*    APPLICATIONS of MICROCOMPUTERS in             *
*        INDEXING                                  *
*                                                  *
```

For procedure on how to set the cutoff threshold value, see Section 10.6.

10.3 Controlling Boundaries Between Headings

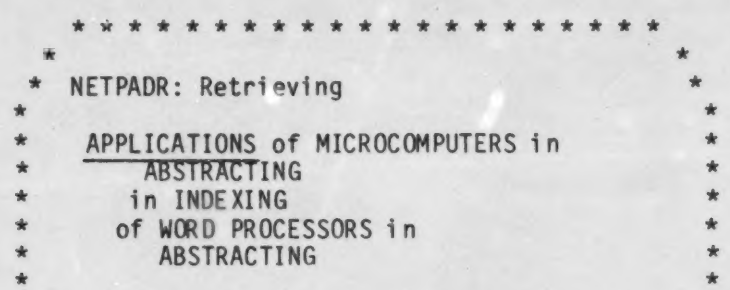
A second value, the subheading threshold, may be used to control boundaries between headings and subheadings, subheadings and subsubheadings, and so on. For example, supposing the database contains three descriptions:

```
NETPADR: Retrieving

 APPLICATIONS of MICROCOMPUTERS in
     ABSTRACTING
 APPLICATIONS of MICROCOMPUTERS in
     INDEXING
 APPLICATIONS of WORD PROCESSORS
     in ABSTRACTING
```

Here each entry consists of a heading only, with no part treated as a subheading. Setting the subheading threshold below the forward weight of the linktype represented by "of" but above that represented by "in" (e.g., to 25) would cause a transition to a subheading wherever a forward link of the latter type occurred, as in:

```
NETPADR: Retrieving

 APPLICATIONS of MICROCOMPUTERS in
     ABSTRACTING
   in INDEXING
 APPLICATIONS of WORD PROCESSORS
     in ABSTRACTING
```

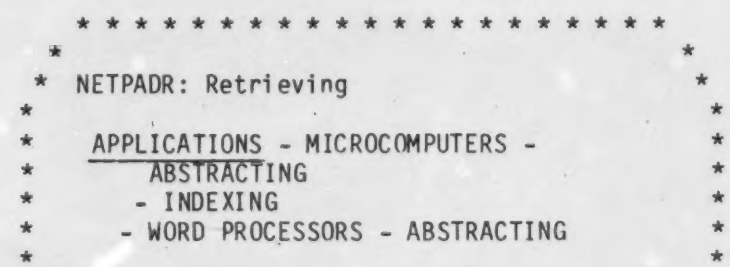
Setting the subheading threshold value above the forward weight of the linktype represented by "of" (e.g., to 35) would yield:



For procedure on how to set the subheading threshold value, see Section 10.6.

10.4 Rewording Connective Expressions

Different indexes adopt different styles of connectives: some make extensive use of natural-language elements such as prepositions; others prefer discreet punctuation to avoid distracting the searcher from the descriptors. The user is thus allowed to modify the connectives associated with any linktype. For example, if a dash were specified as the forward connective for both linktypes, the previous example would appear as:



10.5 Resetting Weights And Connectives

Weights and connectives are reset by editing the appropriate linktype. For procedure see Section 3.3.3.

10.6 Setting Subheading And Cutoff Threshold Values

The procedure involved in setting or resetting subheading and cutoff threshold values is similar to the procedure involved in editing linktypes.

As you might have already noticed, subheading and cutoff threshold values are part of the linktype list. They are the first items in the list. Accordingly. if the cursor is positioned next to that entry and the "!" key is pressed a series of prompts will be generated; as follows:

```
OLD THRESHOLDS:
 SUBHEADING = 0
 BREAKOFF = 0
NEW SUBHEADING THRESHOLD =
NEW BREAKOFF THRESHOLD =
```

The new values are entered following each prompt and each must be followed by a carriage RETURN.

Note that unlike the linktypes, the subheading and cutoff threshold values are not permanently stored in the database. These values must be re-entered at the beginning of each session.

11.0 MAINTAINING THE INVERTED-INDEX FILE

The user is asked to keep track of the number of subnetworks added to the database. Every 15 or 20 subnetworks added the user should call-up the MAINTAINING INDEX module in order to make searching of the inverted-index file more efficient.

The MAINTAINING is module is accessed by:

(1) pressing the "OFF/RVS" key, and

(2) following the prompt:

```
NETPADO:  Choosing
OPTION=?
```

typing in the command MAINTAIN (plus RETURN).

As the module performs its function, the track and sector numbers of the diskette blocks which make up the inverted-index file will one after the other be displayed on the screen; as follows:

```
* * * * * * * * * * * * * * * * * * * * * *
*                                           *
*  NETPADM: Maintaining index               *
*                                           *
*    1        4        1:                   *
*    2        14       GS                   *
*    2        9        IO                   *
*    2        6        SF                   *
*    2        3                             *
*                                           *
*   SAVE?                                   *
*                                           *
```

Following the completion of this procedure the system will ask the user if he/she considers the operation successful via the prompt:

SAVE?

The user can reply either "Y" or "N". If the letter "N" is typed the existing inverted-index file will remain unchanged. If the letter "Y" is typed the newly structured inverted-index file will be accepted. In either case, system command will be returned to the MAIN EDITING module.

12.0 PRODUCING HARD COPIES

Hard copies of the contents of the screen may be produced by pressing the "." key.

> The "." command causes the contents of the current screen to be printed on the printer. It leaves the screen unchanged.

Only one screen may be printed at a time. If we need to print more than one screen we must call and/or generate one screen, print it, call and/or generate the next screen, print it, and so on until all the needed screens are printed.

Please also note that this command is restricted to the following NETPAD modules:

> the MAIN EDITING module, thereby alowing the printing of subnetworks in the "staircase" fashion;
>
> the TABLE DISPLAY module, thereby allowing the printing of subnetworks in table form;
>
> the REDEFINING module, thereby allowing the printing of the linktype list;
>
> the PERMUTING module, thereby allowing the printing of the permuted index entries for the current subnetwork;
>
> the RETRIEVING module, thereby allowing the printing of permuted index displays.

13.0 STANDARDIZING INDEX TERMS

The terms or phrases representing concept nodes with a NETPAD database may be efficiently standardized via the "=" command.

> The "=" command automatically substitutes all the occurrences of a specified term in a NETPAD database with another specified term.

Once the "=" key has been pressed, system command will be transferred to the CHANGING module and the following prompt will be generated:

```
* * * * * * * * * * * * * * * * * * * * * * *
*                                             *
*  NETPADC: Changing                          *
*                                             *
*  OLD TERM=?                                 *
*                                             *
```

At that stage, the user may type in the term or phrase needing substitution. This entry should be followed by a carriage RETURN. In response, the CHANGING module will prompt the user for the term to be used in the substitution; as follows:

```
* * * * * * * * * * * * * * * * * * * * * * *
*                                             *
*  NETPADC: Changing                          *
*                                             *
*  OLD TERM=? CATALOGING                      *
*  NEW TERM=?                                 *
*                                             *
```

Once that term is typed in and followed by a carriage RETURN, the CHANGING module will proceed substituting all the occurences of the "OLD TERM" with the current NETPAD database with the specified "NEW TERM". As it does so, the module will display on the screen both the track and sector numbers of the diskette blocks where a change has occurred; as follows:

```
* * * * * * * * * * * * * * * * * * * * * * *
*                                             *
*  NETPADC: Changing                          *
*                                             *
*  OLD TERM=? CATALOGING                      *
*  NEW TERM=? CATALOGUING                     *
*  SECTOR 5     TRACK 2     CHANGED           *
*  SECTOR 4     TRACK 3     CHANGED           *
*  SECTOR 6     TRACK 5     CHANGED           *
*  SECTOR 1     TRACK 7     CHANGED           *
*  DONE                                       *
*                                             *
```

When all the substitutions have been performed the message "DONE" will briefly appear on the screen, and system command will be returned to the MAIN EDITING module.

14.0 GENERATING FACET PRINTOUT

If desired, an alphabetical list of the terms representing the concept nodes linked with a specified linktype within a NETPAD database may be generated. The command used for this purpose is the ":" command.

> The ":" command automatically triggers the selection, sorting, and printing of the terms or phrases which represent concept nodes linked by a specified linktype.

Once the ":" key is pressed, system command is transferred to the FACET PRINTOUT module and the user is prompted for a linktype mnemonic preceded by either the forward linktype indicator "/" or the backward linktype indicator "\"; as follows:

```
  * * * * * * * * * * * * * * * * * * * * * *
 *                                           *
*  NETPADF: Facet printout                   *
*                                             *
*  PRESS / OR \ PLUS LINKTYPE MNEMONIC:       *
*                                             *
```

At that stage, the user must type either the "/" indicator or the "\" indicator, plus one of the linktype mnemonics used within the database.

In response, the FACET PRINTOUT module will retrieve all the terms or phrases representing the concept nodes which have been linked with the specified linktype and display them in alphabetical order on the Commodore printer; for example:

```
 7          . subject              .
----------------------------------------
FEASIBILITY STUDY
REPORT
STATISTICAL ANALYSIS
SURVEY
SYSTEM ANALYSIS
```

or,

9 of .

APPLICATIONS
CIRCULATION
CLASSIFICATION
MANAGEMENT
PLANNING

15.0 USING INDEXER PRODUCTIVITY AIDS

Although made as simple as possible, inputting concept nodes and assigning links between concept nodes can be very time consuming. In order to reduce the amount of time and effort required to construct subnetworks a user can elect to construct subnetworks using one of three indexer productivity aids. These include:

(1) the syntactic analysis of phrases,

(2) the automatic extraction of concept nodes, and

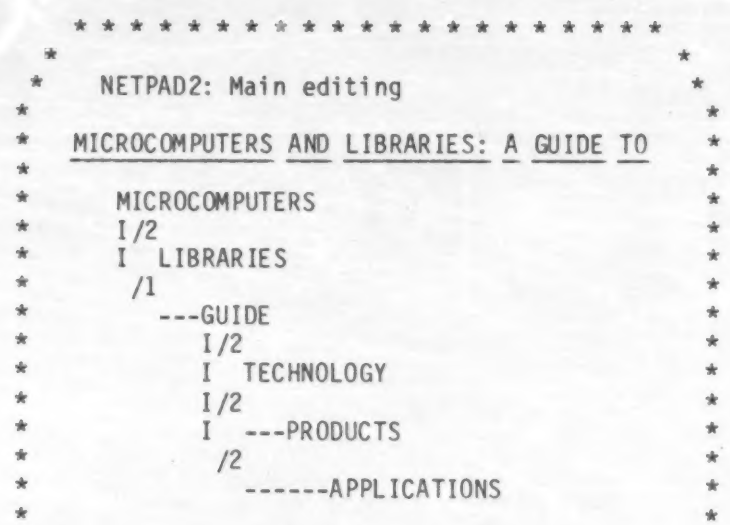
(3) the use of subschemas.

15.1 Syntactic Analysis Of Phrases

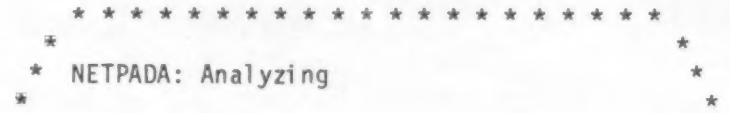
This productivity aid allows the user to syntactically analyse a phrase entered as a concept node. The analysis is automatic and generates a set of syntactically derived concept nodes and links using (1) system defined syntactic rules, and (2) system-defined linktypes. The command used to activate this analysis is the "SHIFTED A" command.

> The "SHIFTED A" command passes system command to the ANALYZING module which in turn syntactically analyses the current node and adds the result of the analysis to the current subnetwork in the form of concept nodes and links.

For example, if the current concept node happened to be the title of a book, e.g., "Microcomputers and libraries: a guide to technology, products and applications" and the "SHIFTED A" key was pressed, the following subnetwork would automatically be generated:



Note that the generation of the above subnetwork will take a few seconds and that during this time the following display will briefly appear on the screen:



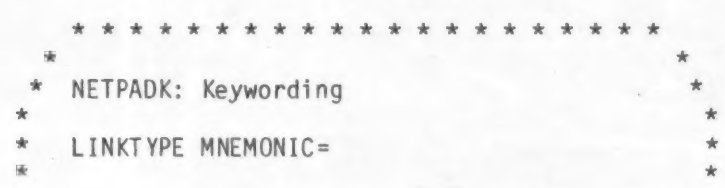
More important however, is to note that the term "AGE", articles, and punctuation marks have been excluded from the result. This was achived partly via a stop word list and partly via a program routine which automatically ignores any word of three letters or less.

15.2 Automatic Extraction Of Index Terms

This productivity aid allows the automatic extraction of index terms from a phrase entered as a concept node. The resulting index terms are added to the current subnetwork in the form of concept nodes either without links or with every concept node linked in the same way to the source node. The command used to activate this extraction is the "SHIFTED K" command.

> The "SHIFTED K" command passes system command to the KEYWORDING module which automatically extracts index terms from the current node and adds the result to the current subnetwork in the form of concept nodes. Links may or may not be generated.

Once the "SHIFTED K" key is pressed, system command is transferred to the KEYWORDING module and the user is prompted for a linktype mnemonic; as follows:



At that stage, one may elect to type in either a valid linktype mnemonic or non-existing linktype mnemonic, such as the letter "x" or "z".

If a valid linktype mnemonic is typed in, the KEYWORDING module will not only extract the index terms from the current concept node; it will also link each extracted term to the source node using the chosen linktype. Such a feature is useful as the resulting concept nodes will be linked in a manner which is conducive to the creation of a KWOC like permuted index; for example:

```
NETPAD2: Main editing

MICROCOMPUTERS AND LIBRARIES: A GUIDE TO
IR/
I  MICROCOMPUTERS
IR/
I  ---LIBRARIES
IR/
I  ------GUIDE
IR/
I  ---------TECHNOLOGY
IR/
I  ------------PRODUCTS
 R/
   --------------APPLICATIONS
```

If a non-existing or invalid linktype mnemonic is typed in, the KEYWORDING module will not assign any links. It will simply extract the index terms from the current concept node and store them in the form of concept nodes. This feature is useful as it allows the user to assign his or her own links, yet benefit from the automatic extraction of concept nodes; for example:

```
NETPAD2: Main editing

MICROCOMPUTERS AND LIBRARIES: A GUIDE TO

   MICROCOMPUTERS

      LIBRARIES

         GUIDE

            TECHNOLOGY

               PRODUCTS

                  APPLICATIONS
```

Note that, as in the ANALYZING module, the term "AGE", articles, and punctuation marks have been excluded from the results. This was achieved in the same as it was achieved in the ANALYSING module, that is, partly via a stop word list and partly via a program routine which automatically ignores any word of three letters or less.

15.3 Using Subschemas

This section is yet to be written.

16.0 EXITING FROM NETPAD

The "STOP/RUN" command finishes the current NETPAD session.

Do not use this command if the light on either diskette drive is lit.

17.0 SUMMARY OF COMMANDS

This section summarizes the various commands you can type in NETPAD. We hope that, long after you are familiar with the functions of NETPAD, you will use this section for reference.

17.1 General Commands

The following commands are general in the sense that none of them is bound to any particular module.

OFF/RVS plus REDEFINE plus RETURN	calls the REDEFINING module
OFF/RVS plus BROWSE plus RETURN	calls the BROWSING module
OFF/RVS plus MAINTAIN plus RETURN	calls the MAINTAIN module
&	calls the QUERYING module
)	calls the STORING module
?	calls the TABLE DISPLAY module
'	calls the SORTING module
%	calls the PERMUTING module
=	calls the CHANGING module
:	calls the FACET PRINTOUT module
SHIFTED A	calls the ANALYSING module
SHIFTED K	calls the KEYWORDING module
.	causes the contents of the current screen to be printed on the printer
STOP/RUN	finishes the current NETPAD session

17.2 Module-bound Commands

The following commands are module-bound in the sense that they can only be used within specific modules. Some of these commands may be used within more than one module, but in that case their meaning is often changed.

`>` in the MAIN EDITING module it moves the reverse-field one node down the "staircase" display

in the REDEFINING module it moves the cursor one position down the linktype list

in the RETRIEVING module it moves the cursor one entry down the list of permutations

in the BROWSING module it transfers into current memory the set of subnetworks addresses attached to the next search term in the inverted-index file

in the GETTING module it moves the cursor one entry down the list of subnetwork addresses

`<` in the MAIN EDITING module it moves the reverse-field one node up the "staircase" display

in the REDEFINING module it moves the cursor one position up the linktype list

in the RETRIEVING module it moves the cursor up the list of permutations

in the GETTING module it moves the cursor one entry up the list of subnetwork addresses

`/` in the MAIN EDITING module it initiates a link from the currently reverse-field node to another note

`\` in the MAIN EDITING module it initiates a link to the currently reverse-field node from another node

`↑` in the MAIN EDITING module it indicates that a link or unlinking operation has been completed

in the RETRIEVING and GETTING modules it indicates that you wish to retrieve the subnetwork pointed to by the cursor

↑ in the BROWSING module it stores the current set of subnetwork addresses as the next set in a set list whose sets are identified by a single letter of the alphabet starting with the letter "A"

← in the MAIN EDITING module it initiates unlinking from the current reverse-field node to some other node

! in the MAIN EDITING module it initiates the editing of the current concept node

in the REDEFINING module it (1) clears linktype n, where n stands for the linktype indicated by the cursor, and (2) triggers a series of prompts which allow the input of linktype information

in the BROWSING module it retrieves the diskette addresses of the subnetworks which have a concept node with a term corresponding to the specified search key

in the MAIN EDITING module it causes the deletion of the current concept node

in the BROWSING module it transfers a previously stored set of subnetwork addresses into current memory

CRSR UP/DOWN

in the MAIN EDITING module it moves the subnetwork "window" two lines up from its current position

in the REDEFINING module it displays the next screenful of linktypes

in the QUERYING and PERMUTING modules it displays the next screenful of permutations

SHIFTED CRSR UP/DOWN

in the MAIN EDITING module it moves the subnetwork "window" two lines down from its current position

CRSR RIGHT/LEFT

in the MAIN EDITING it moves the subnetwork "window" 10 characters left from its current position

SHIFTED CRSR RIGHT/LEFT	in the MAIN EDITING module it moves the subnetwork "window" 10 characters right from its current position
CLR/HOME	in the REDEFINING module it displays the first screenful of linktypes in the QUERYING and PERMUTING modules it displays the first screenful of permutations
SHIFTED CLR/HOME	in the MAIN EDITING module it clears the current subnetwork from the workspace in the BROWSE module it clears the screen, erases the set list and the current set, and re-displays the module command summary at the top of the screen
SHIFTED ↑	in the QUERYING and BROWSING modules it transfers system command back to the MAIN EDITING module
SHIFTED G	in the BROWSE module it transfers system execution to the GETTING module which in turn displays the diskette track and sector numbers of the subnetworks listed in the current set

*n in the BROWSING module it combines two sets using the Boolean operator AND

+n in the BROWSING module it combines two sets using the Boolean operator OR

-n in the BROWSING module it combines two sets using the Boolean operator BUT NOT

17.3 Additional Commands

The following commands have not been explained elsewhere in this manual. Most are experimental and none are vital to the production of index displays using NETPAD.

$	compresses the display
@	converts the current node term to lower case
[	calls the file-read module
]	calls the file-write module
SHIFTED E	calls the extraction module (to make a list of all subnetworks containing the current subnetwork
SHIFTED J	generates additional links of type 5 wherever in the subnetwork there are links of type 4
SHIFTED T	produces tree animation of the current subnetwork
SHIFTED X	produces angular animation of the current subnetwork